

Design Assignment 1

Student Name: Ernesto Ibarra-Ayala

Student #: 2001211571

Student Email: ibarre3@unlv.nevada.edu

Primary Github address: <https://github.com/ErnestoIbarra333>

Directory: <https://github.com/ErnestoIbarra333/ErnestoIbarra.git>

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

So far, we only used Atmel Studios and nothing else just yet. We will be using the atmega328p board soon.

2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A

No initial code given

3. DEVELOPED MODIFIED CODE OF TASK 1/A/B/C/D/E

Here is my code screenshot as well as the actual code copied and pasted. I also put a screenshot of my code building successfully.

DA1 - Microchip Studio Advanced Mode Quick Launch (Ctrl)

File Edit View VAssistX ASF Project Build Debug Tools Window Help

Debug Debug Browser ATmega328P Simulator

main.asm

```

; Created: 6/15/2021 7:17:38 PM
; Author : Ernesto Ibarra
;
;///////////////// A).
.org 0
.def numAH = R16
.def numAL = R17
    LDI numAH, 0x12
    LDI numAL, 0x34
    STS 0x402, numAH
    STS 0x403, numAL

;///////////////// B).
.org 0x998 ;[0x1330]
MYDATA: .db 0x56,0x78

;///////////////// C).
    LDI ZL, LOW(2*MYDATA)
    LDI ZH, HIGH(2*MYDATA)
    LPM R4, Z+ //load values into R4 and R5
    LPM R5, Z
    ADD numAL, R5 // 0x1234 + 0x5678 = 0x68AC
    ADC numAH, R4 // store values in R16 and R17
    LDI YH, HIGH(0x0000)
    LDI YL, LOW(0x0000)
    CALL STORE_IN_EEPROM //stores R16(68) in starting EEPROM starting location
    MOV numAH, numAL
    INC YL
    CALL STORE_IN_EEPROM // stores R17(AC) in the next EEPROM location
    NOP

END: JMP END
;///////////////// D).
.ORG 0x1000 ; 0x100 << 1 = 0x2000 which is starting location wanted
MYDATA1: .dw 0x9100,0x9101,0x9102,0x9103,0x9104,0x9105,0x9106,0x9107,0x9108,0x9109,0x9110
// I tried getting it working with the X pointer but I could not get "SPM" working, also it keeps telling me it is only allowed
// with the Z pointer so I'm not sure what to do here besides doing it the same way we did it in part B. hope this is okay for part D
// thank you.

STORE_IN_EEPROM:
    SBIC EECR, EEPE
    RJMP STORE_IN_EEPROM
    OUT EEARH, YH
    OUT EEARL, YL
    OUT EEDR, numAH
    SBI EECR, EEMPE
    SBI EECR, EEPE
    RET

```

80 %

Error List

Solution Explorer

Search Solution

Solution Explorer

DA

Available...

Properties

```

main.asm
;
; DA1.asm
;
; Created: 6/15/2021 7:17:38 PM
; Author : Ernesto Ibarra
;
////////// A).
.org 0
.def numAH = R16
.def numAL = R17
    LDI numAH, 0x12
    LDI numAL, 0x34
    STS 0x402, numAH
    STS 0x403, numAL

////////// B).
.org 0x998 ;[0x1330]
MYDATA: .db 0x56,0x78

////////// C).
    LDI ZL, LOW(2*MYDATA)
    LDI ZH, HIGH(2*MYDATA)
    LPM R4, Z+ //load values into R4 and R5
    LPM R5, Z
    ADD numAL, R5 // 0x1234 + 0x5678 = 0x68AC
    ADC numAH, R4 // store values in R16 and R17
    LDI YH, HIGH(0x0000)
    LDI YL, LOW(0x0000)
    CALL STORE_IN_EEPROM //stores R16(68) in starting EEPROM starting location
    MOV numAH, numAL
    INC YL
    CALL STORE_IN_EEPROM // stores R17(AC) in the next EEPROM location
    NOP

END: JMP END

////////// D).
.ORG 0x1000 ; 0x1000 << 1 = 0x2000 which is starting location wanted
MYDATA1: .dw 0x9100,0x9101,0x9102,0x9103,0x9104,0x9105,0x9106,0x9107,0x9108,0x9109,0x9110
// I tried getting it working with the X pointer but I could not get "SPM" working, also it keeps telling me it is only allowed
// with the Z pointer so I'm not sure what to do here besides doing it the same way we did it in part B. hope this is okay for part D
// thank you.

```

Output

Show output from: Build

```

[.cseg] 0x000000 0x002026      62      24      86 32768 0.3%
[.dseg] 0x000100 0x000100       0       0       0  2048 0.0%
[.eseg] 0x000000 0x000000       0       0       0  1024 0.0%
Assembly complete, 0 errors, 0 warnings
Done executing task "RunAssemblerTask".
Done building target "CoreBuild" in project "DA1.asmproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "E:\7.0\Vs\Avr.common.targets" from project "C:\Users\Doradoboy\Documents\Atmel Studio\7.0\DA1\DA1\DA1.asmproj" (entry point):
Done building target "Build" in project "DA1.asmproj".
Done building project "DA1.asmproj".

Build succeeded.
===== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====

```

```
////////// A).
```

```
.org 0
.def numAH = R16
.def numAL = R17
    LDI numAH, 0x12
    LDI numAL, 0x34
    STS 0x402, numAH
    STS 0x403, numAL
```

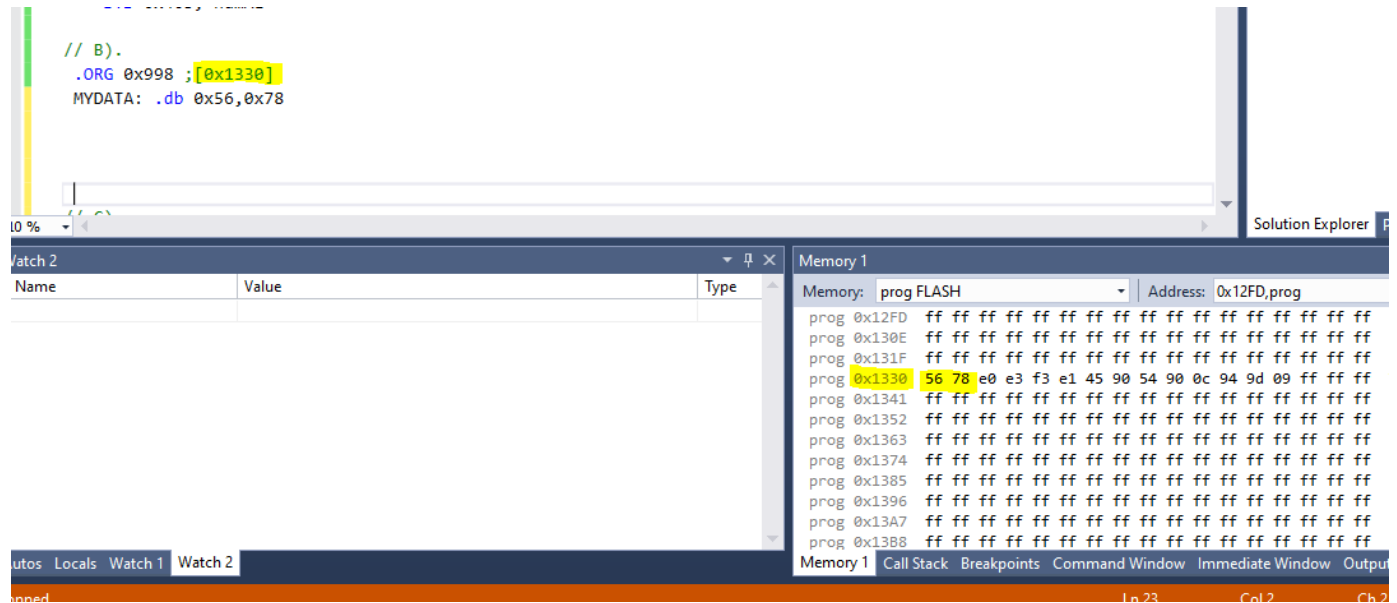
```
////////// B).
```

```
.org 0x998 ;[0x1330]
MYDATA: .db 0x56,0x78
```

```
////////// C).
```

```
    LDI ZL, LOW(2*MYDATA)
    LDI ZH, HIGH(2*MYDATA)
    LPM R4, Z+ //load values into R4 and R5
    LPM R5, Z
    ADD numAL, R5 // 0x1234 + 0x5678 = 0x68AC
    ADC numAH, R4 // store values in R16 and R17
    LDI YH, HIGH(0x0000)
    LDI YL, LOW(0x0000)
```


1B). Here is part B



6. 1C). Here is part C, I mostly commented the entire code so It would be easier to see the comments right next to the code instead of commenting it here. Hopes it's ok.

Registers

R00 = 0x00 R01 = 0x00 R02 = 0x00 R03 = 0x00 R04 = 0x56 R05 = 0x78 R06 = 0x00 R07 = 0x00 R08 = 0x00 R09 = 0x00 R10 = 0x00 R11 = 0x00
R12 = 0x00 R13 = 0x00 R14 = 0x00 R15 = 0x00 R16 = 0xAC R17 = 0xAC R18 = 0x00 R19 = 0x00 R20 = 0x00 R21 = 0x00 R22 = 0x00 R23 = 0x00
R24 = 0x00 R25 = 0x00 R26 = 0x00 R27 = 0x00 R28 = 0x01 R29 = 0x00 R30 = 0x31 R31 = 0x13

main.asm

```

//////////////// B).
.ORG 0x998 ;[0x1330]
MYDATA: .db 0x56,0x78
//////////////// C).
LDI ZL, LOW(2*MYDATA)
LDI ZH, HIGH(2*MYDATA)
LPM R4, Z+ //load values into R4 and R5
LPM R5, Z
ADD numAL, R5 // 0x1234 + 0x5678 = 0x68AC
ADC numAH, R4 // store values in R16 and R17
LDI YH, HIGH(0x0000)
LDI YL, LOW(0x0000)
CALL STORE_IN_EEPROM //stores R16(68) in starting EEPROM starting location
MOV numAH, numAL
INC YL
CALL STORE_IN_EEPROM // stores R17(AC) in the next EEPROM location
NOP
END: JMP END
STORE_IN_EEPROM:
SBIC EECR, EEPE
RJMP STORE_IN_EEPROM
OUT EEARH, YH
OUT EEARL, YL
OUT EEDR, numAH
SBI EECR, EEMPE
SBI EECR, EEPE
RET

```

100 %

Watch 2

Name	Value	Type
------	-------	------

Memory 1

Memory: eeprom EEPROM Address: 0x0000,eeprom

eeprom 0x0000	68 ac ff ff ff ff ff ff ff ff ff ff ff ff ff ff
eeprom 0x0011	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
eeprom 0x0022	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
eeprom 0x0033	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
eeprom 0x0044	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
eeprom 0x0055	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
eeprom 0x0066	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
eeprom 0x0077	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
eeprom 0x0088	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
eeprom 0x0099	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
eeprom 0x00AA	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
eeprom 0x00BB	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

Autos Locals Watch 1 Watch 2

Memory 1 Call Stack Breakpoints Command Window Immediate Wind

Stopped Ln 34 Col 1

7. **1D).** Here is part D, I tried getting it working with the X pointer but I could not get "SPM" working, also it keeps telling me it is only allowed with the Z pointer so I'm not sure what to do here besides doing it the same way we did it in part B. hope this is okay for part D thank you.

R00 = 0x00 R01 = 0x00 R02 = 0x00 R03 = 0x00 R04 = 0x56 R05 = 0x78 R06 = 0x00 R07 = 0x00 R08 = 0x00 R09 = 0x00 R10 = 0x00 R11 = 0x00
R12 = 0x00 R13 = 0x00 R14 = 0x00 R15 = 0x00 R16 = 0x56 R17 = 0xAC R18 = 0x00 R19 = 0x00 R20 = 0x00 R21 = 0x00 R22 = 0x00 R23 = 0x00
R24 = 0x00 R25 = 0x00 R26 = 0x00 R27 = 0x00 R28 = 0x01 R29 = 0x00 R30 = 0x30 R31 = 0x13

```

main.asm
CALL STORE_IN_EEPROM //stores R16(68) in starting EEPROM starting location
MOV numAH, numAL
INC YL
CALL STORE_IN_EEPROM // stores R17(AC) in the next EEPROM location
NOP

////////// D).
.ORG 0x1000 ; 0x100 << 1 = 0x200 which is starting location wanted
MYDATA1: .dw 0x9100,0x9101,0x9102,0x9103,0x9104,0x9105,0x9106,0x9107,0x9108,0x9109,0x9110

END: JMP END
STORE_IN_EEPROM:
SBIC EECR, EEPE
RJMP STORE_IN_EEPROM
OUT EEARH, YH
OUT EEARL, YL
OUT EEDR, numAH
SBI EECR, EEPE
SBI EECR, EEPE
RET

```

100 %

Name	Value	Type
Watch 2		

Memory	Address: 0x1FCA,prog
prog 0x1FCA	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
prog 0x1FDC	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
prog 0x1FEE	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
prog 0x2000	00 91 01 91 02 91 03 91 04 91 05 91 06 91 07 91 08 91
prog 0x2012	09 91 10 91 0c 94 0b 10 f9 99 fe cf d2 bd c1 bd 00 bd
prog 0x2024	fa 9a f9 9a 08 95 ff ff ff ff ff ff ff ff ff ff ff
prog 0x2036	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
prog 0x2048	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
prog 0x205A	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
prog 0x206C	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
prog 0x207E	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

8. **1E).** Here is part E, the program will take approximately 514us to execute since we have about 8230 cycles in this program.

Registers

R00 = 0x00 R01 = 0x00 R02 = 0x00 R03 = 0x00 R04 = 0x56 R05 = 0x78 R06 = 0x00 R07 = 0x00 R08 = 0x00 R09 = 0x00 R10 = 0x00 R11 = 0x00
R12 = 0x00 R13 = 0x00 R14 = 0x00 R15 = 0x00 R16 = 0x56 R17 = 0xAC R18 = 0x00 R19 = 0x00 R20 = 0x00 R21 = 0x00 R22 = 0x00 R23 = 0x00
R24 = 0x00 R25 = 0x00 R26 = 0x00 R27 = 0x00 R28 = 0x01 R29 = 0x00 R30 = 0x30 R31 = 0x13

Processor Status - X main.asm

Name	Value
Program Counter	0x0001005
Stack Pointer	0x08FF
X Register	0x0000
Y Register	0x0001
Z Register	0x1330
Status Register	00000000
Cycle Counter	8230
Frequency	1.000 MHz
Stop Watch	8,230.00 µs
Registers	
R00	0x00
R01	0x00
R02	0x00
R03	0x00
R04	0x56
R05	0x78
R06	0x00
Watch 2	
Name	Value

Untitled - Paint

File Home View

Paste Cut Copy Select Crop Resize Rotate Brushes

Clipboard Image Tools Shapes

Outline Fill Size Color 1 Color 2

$8230 \times 0.0625 \mu s = 514 \mu s$

9. SCREENSHOT OF EACH DEMO (BOARD SETUP)

10. VIDEO LINKS OF EACH DEMO

11. GITHUB LINK OF THIS DA

<https://github.com/Ernestolbarra333/Ernestolbarra.git>

Student Academic Misconduct Policy

<http://studentconduct.unlv.edu/misconduct/policy.html>

"This assignment submission is my own, original work".

NAME OF THE STUDENT